

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently amended): A method for parallelizing a computer application program based on a script of a script-driven software tool, comprising automatically analyzing the script and producing a parallel computation specification based on such analysis, where such parallel computation specification provides functional equivalence to the script when executed by a parallel runtime system, by:
- (a) parsing the script into statements comprising at least processing steps and dataset definitions;
 - (b) constructing a serial dataflow graph from the parsed statements, the serial dataflow graph having nodes connected by directed edges, the nodes representing datasets, processing steps, and intermediate results; and
 - (c) constructing a parallel dataflow graph from the nodes of the serial dataflow graph such that the parallel dataflow graph may be executed by a parallel runtime system.
2. (Currently amended): A method for parallelizing a computer application program based on a script of a script-driven software tool, comprising automatically analyzing the script and producing a parallel computation specification plus a script fragment set based on such analysis, where such parallel computation specification and script fragment set provides functional equivalence to the script when executed by a parallel runtime system, by:
- (a) parsing the script into statements comprising at least processing steps and dataset definitions;
 - (b) constructing a serial dataflow graph from the parsed statements, the serial dataflow graph having nodes connected by directed edges, the nodes representing datasets, processing steps, and intermediate results; and

- (c) constructing a parallel dataflow graph from the nodes of the serial dataflow graph such that the parallel dataflow graph may be executed by a parallel runtime system; and
- (d) analyzing the parallel dataflow graph to generate script fragments in a form that enables the script-driven software tool to execute some of the processing steps.

3. (Canceled)

4. (Previously presented): The method of claims 1 or 2, wherein constructing the serial dataflow graph includes:

- (a) constructing a serial dataset table of datasets used by the script;
- (b) constructing a serial processing step table of statements performed by the script; and
- (c) constructing a serial dataset access table indicating datasets in the dataset table used by statements in the processing step table.

5. (Previously presented): The method of claims 1 or 2, wherein constructing the parallel dataflow graph includes:

- (a) constructing a parallel dataset table of datasets based on the serial dataset table;
- (b) constructing a parallel processing step table of statements based on the serial processing step table;
- (c) constructing a dataset access table based on the serial dataset access table; and
- (d) determining, for each processing step identified in the parallel processing step table, if a corresponding pre-defined parallelization rewrite rule exists for such processing step, and if so, then applying the corresponding pre-defined parallelization rewrite rule to redefine associated entries in the parallel dataset table, the parallel processing step table, and the dataset access table as parallel processing entries; and if not, then defining such associated entries as serial processing entries.

6. (Original): The method of claim 5, further including resolving any existing partitioning conflicts in the constructed parallel dataflow graph.

7. (Original): The method of claim 5, wherein at least one pre-defined parallelization rewrite rule is an algorithm selected from the group comprising simple partitioning, key-based partitioning, local-global division, external parallelism algorithm, and statement decomposition.

8. (Original): The method of claims 1 or 2, wherein the script-driven software tool is SAS®.


9. (Original): The method of claims 1 or 2, wherein producing the parallel computation specification includes applying at least one pre-defined parallelization rewrite algorithm selected from the group comprising simple partitioning, key-based partitioning, local-global division, external parallelism algorithm, and statement decomposition.

10. (Currently amended): A computer program, residing on a computer-readable medium, for parallelizing a computer application program based on a script of a script-driven software tool, the computer program comprising instructions for causing a computer to automatically analyze the script and produce a parallel computation specification based on such analysis, where such parallel computation specification provides functional equivalence to the script when executed by a parallel runtime system, by:

- (a) parsing the script into statements comprising at least processing steps and dataset definitions;
- (b) constructing a serial dataflow graph from the parsed statements, the serial dataflow graph having nodes connected by directed edges, the nodes representing datasets, processing steps, and intermediate results; and
- (c) constructing a parallel dataflow graph from the nodes of the serial dataflow graph such that the parallel dataflow graph may be executed by a parallel runtime system.

11. (Currently amended): A computer program, residing on a computer-readable medium, for parallelizing a computer application program based on a script of a script-driven software tool, the computer program comprising instructions for causing a computer to automatically

analyze the script and produce a parallel computation specification plus a script fragment set based on such analysis, where such parallel computation specification and script fragment set provides functional equivalence to the script when executed by a parallel runtime system, by:

- 
- (a) parsing the script into statements comprising at least processing steps and dataset definitions;
 - (b) constructing a serial dataflow graph from the parsed statements, the serial dataflow graph having nodes connected by directed edges, the nodes representing datasets, processing steps, and intermediate results; and
 - (c) constructing a parallel dataflow graph from the nodes of the serial dataflow graph such that the parallel dataflow graph may be executed by a parallel runtime system; and
 - (d) analyzing the parallel dataflow graph to generate script fragments in a form that enables the script-driven software tool to execute some of the processing steps.

12. (Canceled)

13. (Previously presented): The computer program of claims 10 or 11, wherein constructing the serial dataflow graph includes:

- (a) constructing a serial dataset table of datasets used by the script;
- (b) constructing a serial processing step table of statements performed by the script; and
- (c) constructing a serial dataset access table indicating datasets in the dataset table used by statements in the processing step table.

14. (Previously presented): The computer program of claims 10 or 11, wherein constructing the parallel dataflow graph includes:

- (a) constructing a parallel dataset table of datasets based on the serial dataset table;
- (b) constructing a parallel processing step table of statements based on the serial processing step table;
- (c) constructing a dataset access table based on the serial dataset access table; and
- (d) determining, for each processing step identified in the parallel processing step table, if a corresponding pre-defined parallelization rewrite rule exists for such processing step,

and if so, then applying the corresponding pre-defined parallelization rewrite rule to redefine associated entries in the parallel dataset table, the parallel processing step table, and the dataset access table as parallel processing entries; and if not, then defining such associated entries as serial processing entries.

15. (Original): The computer program of claim 14, further including resolving any existing partitioning conflicts in the constructed parallel dataflow graph.
16. (Original): The computer program of claim 14, wherein at least one pre-defined parallelization rewrite rule is an algorithm selected from the group comprising simple partitioning, key-based partitioning, local-global division, external parallelism algorithm, and statement decomposition.
17. (Original): The computer program of claims 10 or 11, wherein the script-driven software tool is SAS®.
18. (Original): The computer program of claims 10 or 11, wherein producing the parallel computation specification includes applying at least one pre-defined parallelization rewrite algorithm selected from the group comprising simple partitioning, key-based partitioning, local-global division, external parallelism algorithm, and statement decomposition.
19. (Currently amended): A system for parallelizing a computer application program based on a script of a script-driven software tool, and for automatically analyzing the script and producing a parallel computation specification based on such analysis, where such parallel computation specification provides functional equivalence to the script when executed by a parallel runtime system, including:
- (a) means for parsing the script into statements comprising at least processing steps and dataset definitions;

- (b) means for constructing a serial dataflow graph from the parsed statements, the serial dataflow graph having nodes connected by directed edges, the nodes representing datasets, processing steps, and intermediate results; and
- (c) means for constructing a parallel dataflow graph from the nodes of the serial dataflow graph such that the parallel dataflow graph may be executed by a parallel runtime system.

20. (Currently amended): A system for parallelizing a computer application program based on a script of a script-driven software tool, and for automatically analyzing the script and producing a parallel computation specification plus a script fragment set based on such analysis, where such parallel computation specification and script fragment set provides functional equivalence to the script when executed by a parallel runtime system, including:

- (a) means for parsing the script into statements comprising at least processing steps and dataset definitions;
- (b) means for constructing a serial dataflow graph from the parsed statements, the serial dataflow graph having nodes connected by directed edges, the nodes representing datasets, processing steps, and intermediate results; and
- (c) means for constructing a parallel dataflow graph from the nodes of the serial dataflow graph such that the parallel dataflow graph may be executed by a parallel runtime system; and
- (d) means for analyzing the parallel dataflow graph to generate script fragments in a form that enables the script-driven software tool to execute some of the processing steps.

21. (Canceled)

22. (Previously presented): The system of claims 19 or 20, wherein the means for constructing the serial dataflow graph includes means for:

- (a) constructing a serial dataset table of datasets used by the script;
- (b) constructing a serial processing step table of statements performed by the script; and

- (c) constructing a serial dataset access table indicating datasets in the dataset table used by statements in the processing step table.

23. (Previously presented): The system of claims 19 or 20, wherein the means for constructing the parallel dataflow graph includes means for:

- (a) constructing a parallel dataset table of datasets based on the serial dataset table;
- (b) constructing a parallel processing step table of statements based on the serial processing step table;
- (c) constructing a dataset access table based on the serial dataset access table; and
- (d) determining, for each processing step identified in the parallel processing step table, if a corresponding pre-defined parallelization rewrite rule exists for such processing step, and if so, then applying the corresponding pre-defined parallelization rewrite rule to redefine associated entries in the parallel dataset table, the parallel processing step table, and the dataset access table as parallel processing entries; and if not, then defining such associated entries as serial processing entries.

24. (Original): The system of claim 23, further including means for resolving any existing partitioning conflicts in the constructed parallel dataflow graph.

25. (Original): The system of claim 23, wherein at least one pre-defined parallelization rewrite rule is an algorithm selected from the group comprising simple partitioning, key-based partitioning, local-global division, external parallelism algorithm, and statement decomposition.

26. (Original): The system of claims 19 or 20, wherein the script-driven software tool is SAS®.

27. (Original): The system of claims 19 or 20, wherein the means for producing the parallel computation specification includes means for applying at least one pre-defined parallelization rewrite algorithm selected from the group comprising simple partitioning, key-based

partitioning, local-global division, external parallelism algorithm, and statement decomposition.

28. (New): A method for parallelizing a computer application program based on a script of a script-driven software tool, comprising automatically analyzing the script and producing a parallel computation specification based on such analysis, where such parallel computation specification provides functional equivalence to the script when executed by a parallel runtime system, by:

(a) parsing the script into statements;

(b) constructing a serial dataflow graph from the parsed statements, said constructing including

(i) constructing a serial dataset table of datasets used by the script,

(ii) constructing a serial processing step table of statements performed by the script, and

(iii) constructing a serial dataset access table indicating datasets in the dataset table used by statements in the processing step table; and

(c) constructing a parallel dataflow graph from the serial dataflow graph.

29. (New): A method for parallelizing a computer application program based on a script of a script-driven software tool, comprising automatically analyzing the script and producing a parallel computation specification plus a script fragment set based on such analysis, where such parallel computation specification and script fragment set provides functional equivalence to the script when executed by a parallel runtime system, by:

(a) parsing the script into statements;

(b) constructing a serial dataflow graph from the parsed statements, said constructing including

(i) constructing a serial dataset table of datasets used by the script;

(ii) constructing a serial processing step table of statements performed by the script; and

- (iii) constructing a serial dataset access table indicating datasets in the dataset table used by statements in the processing step table; and
- (c) constructing a parallel dataflow graph from the serial dataflow graph.

30. (New): A method for parallelizing a computer application program based on a script of a script-driven software tool, comprising automatically analyzing the script and producing a parallel computation specification based on such analysis, where such parallel computation specification provides functional equivalence to the script when executed by a parallel runtime system, by:

- (a) parsing the script into statements;
- (b) constructing a serial dataflow graph from the parsed statements, said constructing including
 - (i) constructing a parallel dataset table of datasets based on the serial dataset table;
 - (ii) constructing a parallel processing step table of statements based on the serial processing step table;
 - (iii) constructing a dataset access table based on the serial dataset access table; and
 - (iv) determining, for each processing step identified in the parallel processing step table, if a corresponding pre-defined parallelization rewrite rule exists for such processing step, and if so, then applying the corresponding pre-defined parallelization rewrite rule to redefine associated entries in the parallel dataset table, the parallel processing step table, and the dataset access table as parallel processing entries; and if not, then defining such associated entries as serial processing entries.; and
- (c) constructing a parallel dataflow graph from the serial dataflow graph.

31. (New): A method for parallelizing a computer application program based on a script of a script-driven software tool, comprising automatically analyzing the script and producing a parallel computation specification plus a script fragment set based on such analysis, where

such parallel computation specification and script fragment set provides functional equivalence to the script when executed by a parallel runtime system, by:

- (a) parsing the script into statements;
- (b) constructing a serial dataflow graph from the parsed statements, said instructing including
- (i) constructing a parallel dataset table of datasets based on the serial dataset table;
 - (ii) constructing a parallel processing step table of statements based on the serial processing step table;
 - (iii) constructing a dataset access table based on the serial dataset access table; and
 - (iv) determining, for each processing step identified in the parallel processing step table, if a corresponding pre-defined parallelization rewrite rule exists for such processing step, and if so, then applying the corresponding pre-defined parallelization rewrite rule to redefine associated entries in the parallel dataset table, the parallel processing step table, and the dataset access table as parallel processing entries; and if not, then defining such associated entries as serial processing entries; and
- (c) constructing a parallel dataflow graph from the serial dataflow graph.

32. (New): A computer program, residing on a computer-readable medium, for parallelizing a computer application program based on a script of a script-driven software tool, the computer program comprising instructions for causing a computer to automatically analyze the script and produce a parallel computation specification based on such analysis, where such parallel computation specification provides functional equivalence to the script when executed by a parallel runtime system, by:

- (a) parsing the script into statements;
- (b) constructing a serial dataflow graph from the parsed statements, said constructing including
- (i) constructing a serial dataset table of datasets used by the script;
 - (ii) constructing a serial processing step table of statements performed by the script; and

- (iii) constructing a serial dataset access table indicating datasets in the dataset table used by statements in the processing step table; and
- (c) constructing a parallel dataflow graph from the serial dataflow graph.

33. (New): A computer program, residing on a computer-readable medium, for parallelizing a computer application program based on a script of a script-driven software tool, the computer program comprising instructions for causing a computer to automatically analyze the script and produce a parallel computation specification plus a script fragment set based on such analysis, where such parallel computation specification and script fragment set provides functional equivalence to the script when executed by a parallel runtime system, by:

- (a) parsing the script into statements;
- (b) constructing a serial dataflow graph from the parsed statements, said constructing including
 - (i) constructing a serial dataset table of datasets used by the script;
 - (ii) constructing a serial processing step table of statements performed by the script; and
 - (iii) constructing a serial dataset access table indicating datasets in the dataset table used by statements in the processing step table; and
- (c) constructing a parallel dataflow graph from the serial dataflow graph.

34. (New): A computer program, residing on a computer-readable medium, for parallelizing a computer application program based on a script of a script-driven software tool, the computer program comprising instructions for causing a computer to automatically analyze the script and produce a parallel computation specification based on such analysis, where such parallel computation specification provides functional equivalence to the script when executed by a parallel runtime system, by:

- (a) parsing the script into statements;
- (b) constructing a serial dataflow graph from the parsed statements, said constructing including

- C1
- (i) constructing a parallel dataset table of datasets based on the serial dataset table;
 - (ii) constructing a parallel processing step table of statements based on the serial processing step table;
 - (iii) constructing a dataset access table based on the serial dataset access table; and
 - (iv) determining, for each processing step identified in the parallel processing step table, if a corresponding pre-defined parallelization rewrite rule exists for such processing step, and if so, then applying the corresponding pre-defined parallelization rewrite rule to redefine associated entries in the parallel dataset table, the parallel processing step table, and the dataset access table as parallel processing entries; and if not, then defining such associated entries as serial processing entries; and
 - (c) constructing a parallel dataflow graph from the serial dataflow graph.

35. (New): A computer program, residing on a computer-readable medium, for parallelizing a computer application program based on a script of a script-driven software tool, the computer program comprising instructions for causing a computer to automatically analyze the script and produce a parallel computation specification plus a script fragment set based on such analysis, where such parallel computation specification and script fragment set provides functional equivalence to the script when executed by a parallel runtime system, by:

- (a) parsing the script into statements;
- (b) constructing a serial dataflow graph from the parsed statements, said constructing including
 - (i) constructing a parallel dataset table of datasets based on the serial dataset table;
 - (ii) constructing a parallel processing step table of statements based on the serial processing step table;
 - (iii) constructing a dataset access table based on the serial dataset access table; and
 - (iv) determining, for each processing step identified in the parallel processing step table, if a corresponding pre-defined parallelization rewrite rule exists for such processing step, and if so, then applying the corresponding pre-defined parallelization rewrite rule to redefine

associated entries in the parallel dataset table, the parallel processing step table, and the dataset access table as parallel processing entries; and if not, then defining such associated entries as serial processing entries; and

(c) constructing a parallel dataflow graph from the serial dataflow graph.

36. (New): A system for parallelizing a computer application program based on a script of a script-driven software tool, comprising means and for automatically analyzing the script and means for producing a parallel computation specification based on such analysis, where such parallel computation specification provides functional equivalence to the script when executed by a parallel runtime system, including:

(a) means for parsing the script into statements;

(b) means for constructing a serial dataflow graph from the parsed statements, said means including means for

(i) constructing a serial dataset table of datasets used by the script;

(ii) constructing a serial processing step table of statements performed by the script; and

(iii) constructing a serial dataset access table indicating datasets in the dataset table used by statements in the processing step table; and

(c) means for constructing a parallel dataflow graph from the serial dataflow graph.

37. (New): A system for parallelizing a computer application program based on a script of a script-driven software tool, comprising means and for automatically analyzing the script and means for producing a parallel computation specification plus a script fragment set based on such analysis, where such parallel computation specification and script fragment set provides functional equivalence to the script when executed by a parallel runtime system, including:

(a) means for parsing the script into statements;

(b) means for constructing a serial dataflow graph from the parsed statements, said means including means for

(i) constructing a serial dataset table of datasets used by the script;

- (ii) constructing a serial processing step table of statements performed by the script; and
- (ii) constructing a serial dataset access table indicating datasets in the dataset table used by statements in the processing step table; and
- (c) means for constructing a parallel dataflow graph from the serial dataflow graph.

38. (New): A system for parallelizing a computer application program based on a script of a script-driven software tool, comprising means and for automatically analyzing the script and means for producing a parallel computation specification based on such analysis, where such parallel computation specification provides functional equivalence to the script when executed by a parallel runtime system, including:

- (a) means for parsing the script into statements;
- (b) means for constructing a serial dataflow graph from the parsed statements, said means including means for
 - (i) constructing a parallel dataset table of datasets based on the serial dataset table;
 - (ii) constructing a parallel processing step table of statements based on the serial processing step table;
 - (iii) constructing a dataset access table based on the serial dataset access table; and
 - (iv) determining, for each processing step identified in the parallel processing step table, if a corresponding pre-defined parallelization rewrite rule exists for such processing step, and if so, then applying the corresponding pre-defined parallelization rewrite rule to redefine associated entries in the parallel dataset table, the parallel processing step table, and the dataset access table as parallel processing entries; and if not, then defining such associated entries as serial processing entries; and
- (c) means for constructing a parallel dataflow graph from the serial dataflow graph.

39. (New): A system for parallelizing a computer application program based on a script of a script-driven software tool, comprising means and for automatically analyzing the script and means for producing a parallel computation specification plus a script fragment set based on

such analysis, where such parallel computation specification and script fragment set provides functional equivalence to the script when executed by a parallel runtime system, including:

(a) means for parsing the script into statements;

(b) means for constructing a serial dataflow graph from the parsed statements, said means including means for

(i) constructing a parallel dataset table of datasets based on the serial dataset table;

(ii) constructing a parallel processing step table of statements based on the serial processing step table;

(iii) constructing a dataset access table based on the serial dataset access table; and

(iv) determining, for each processing step identified in the parallel processing step table, if a corresponding pre-defined parallelization rewrite rule exists for such processing step, and if so, then applying the corresponding pre-defined parallelization rewrite rule to redefine associated entries in the parallel dataset table, the parallel processing step table, and the dataset access table as parallel processing entries; and if not, then defining such associated entries as serial processing entries; and

(c) means for constructing a parallel dataflow graph from the serial dataflow graph.
